SLAC NATIONAL ACCELERATOR LABORATORY

FPGA Based BiSS C Splitter

Austin Gnecco[1], Tyler Johnson[2+]

[1]LCLS Intern 2020

[2]Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.

[+]Contact: tjohnson@slac.stanford.edu

Introduction

The problem that I was tasked with was to find a way to record the information from a closed loop encoder system, so that it could be monitored by the greater LCLS beamline control system. Currently, there are extremely precise encoders on the LCLS mirror tables, however, they only communicate with their attached motor, and do not offer a native option to record position data.

The proposed solution was to design a FPGA based signal splitter to be used with the encoders on the mirror tables at LCLS. The job of this signal splitter is to take the position information from the encoder and (upon request) deliver that information to either the control motor or a data acquisition device to log its position or report it to the greater control system.

I approached this design by first creating a simulation in MATLAB, and then using that simulation to generate the VHDL code that will be used to run the splitter. The VHDL code will then be applied to a Artix-7 development board, to split the signals.

Design

Many potential designs were created, but eventually the one that was settled on used two banks of memory that would be switched between to provide the most accurate position information. The design works as such:

The encoder and motor use a protocol called BiSS-C in order to communicate (Fig. 1). In this protocol, a signal is sent on the MA line from the commanding device to act as a clock. It is a square wave that can range anywhere between 1mhz and 10mhz depending on the device. As not every device is required to use the same frequency, the splitter had to be capable of dealing with multiple clock speeds at the same time.

All interactions with the encoder are controlled through the BiSS-C Master Module(Fig. 3). This module continuously sends a 10 MHz square wave to the encoder, in order to request new location data as often as possible. As the information is received back from the encoder, this module also counts the number of bits received and looks for errors before passing the information on to the buffer module.

Upon the encoder signals entering the Buffer Module (Fig. 4), the Task Scheduler (Fig. 5) looks at all of the different signals coming into the FPGA and decides which requests will be executed first. The DAQ and Motor BiSS

MATLAB Design

Fig.1 – BiSS C Waveform

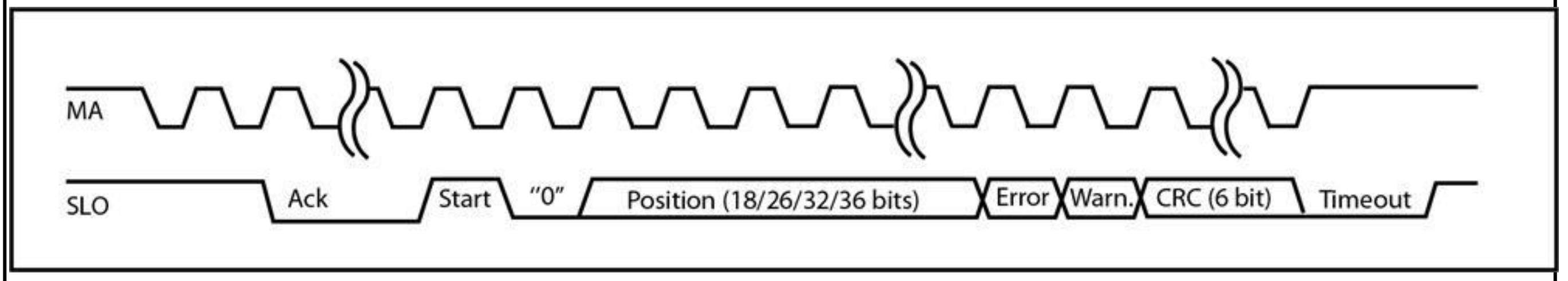


Fig. 2 – System Overview

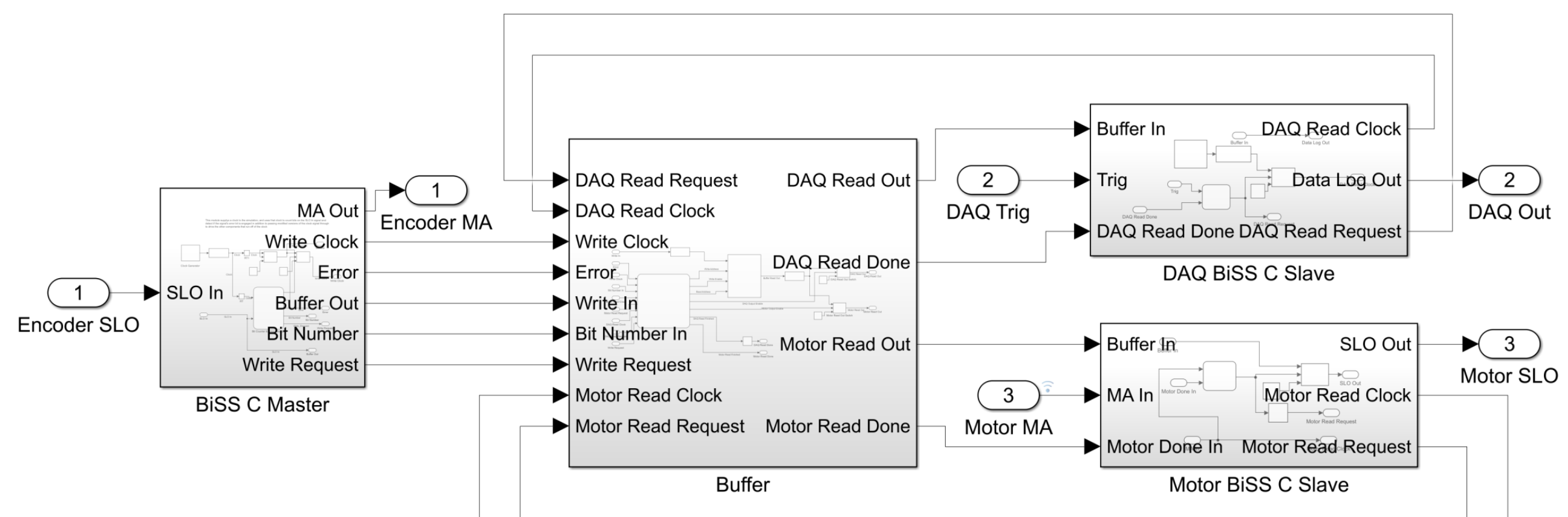


Fig. 3 – BiSS C Master Module

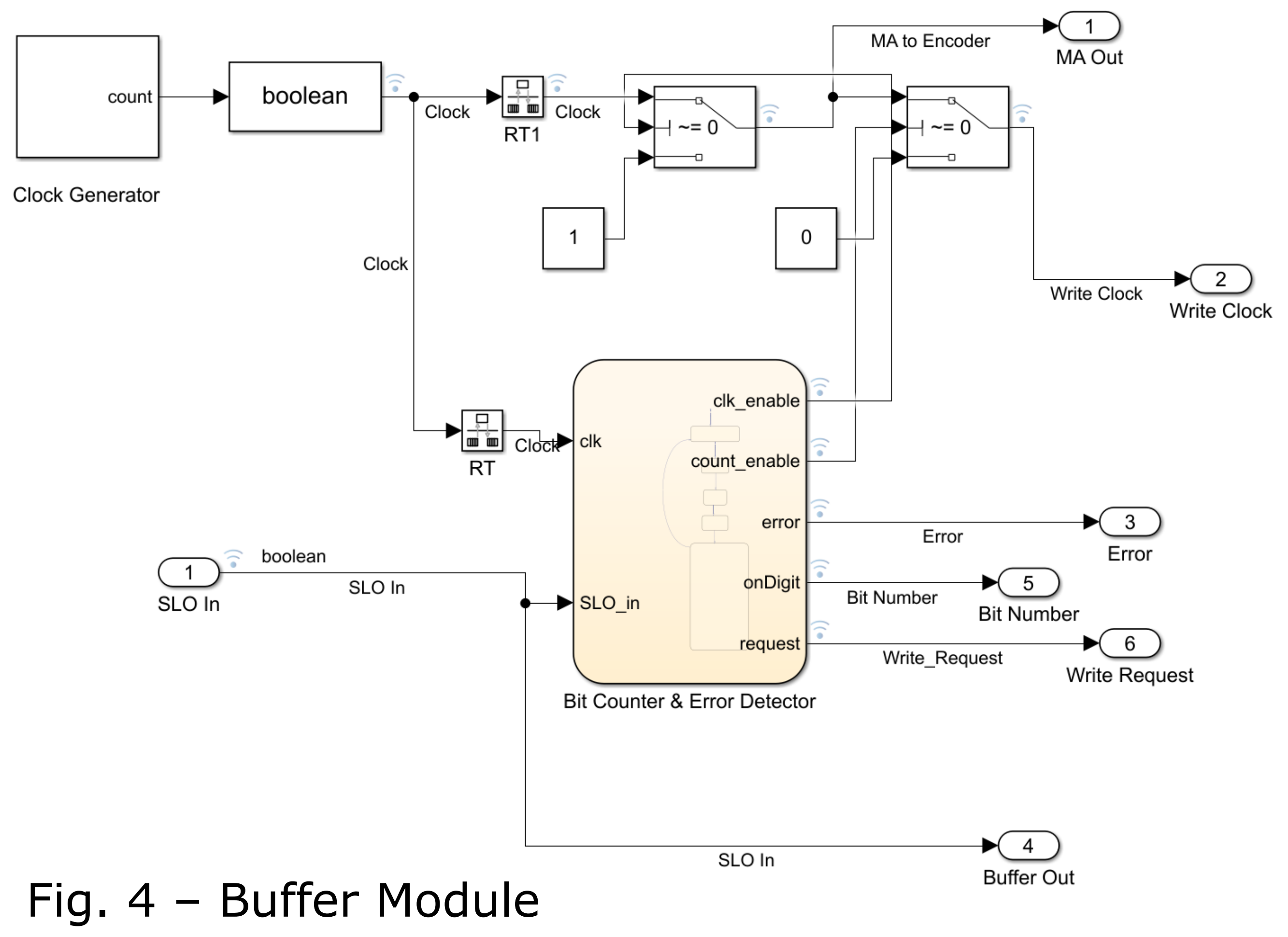


Fig. 4 – Buffer Module

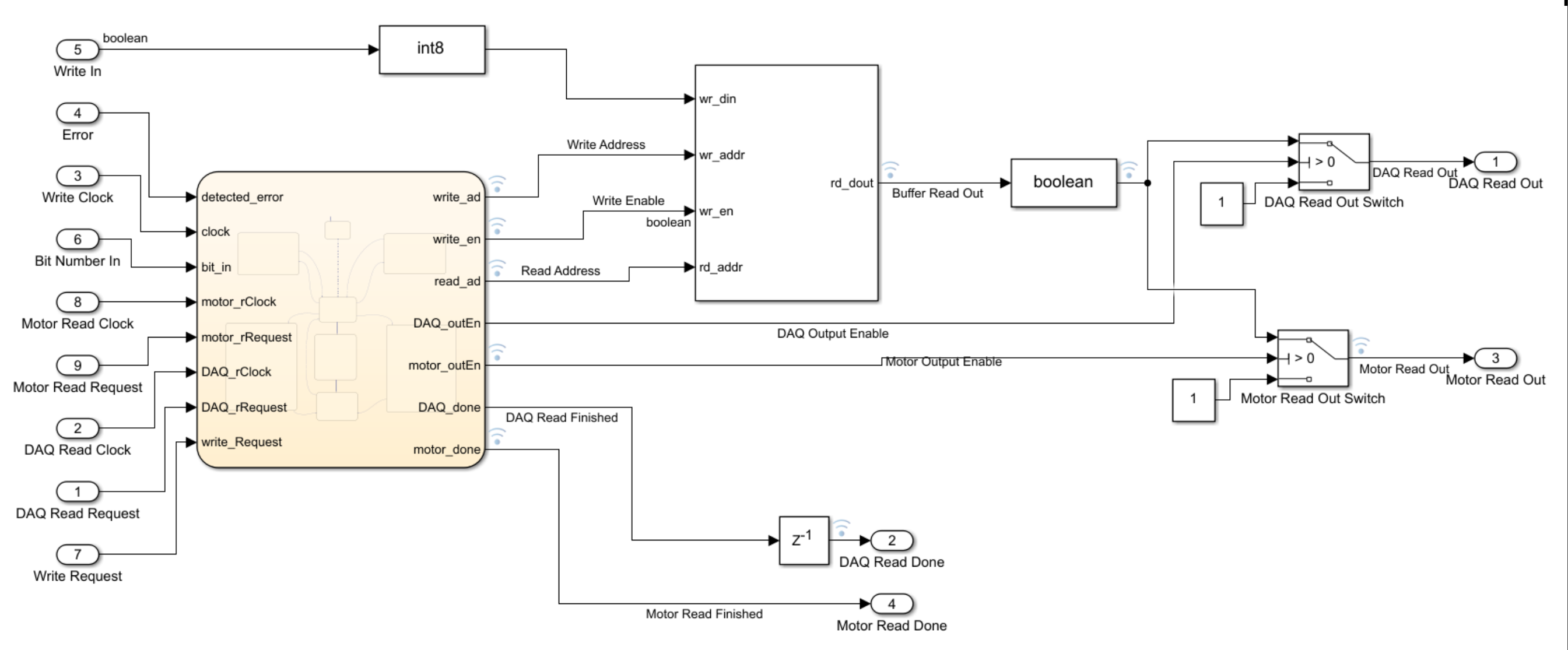
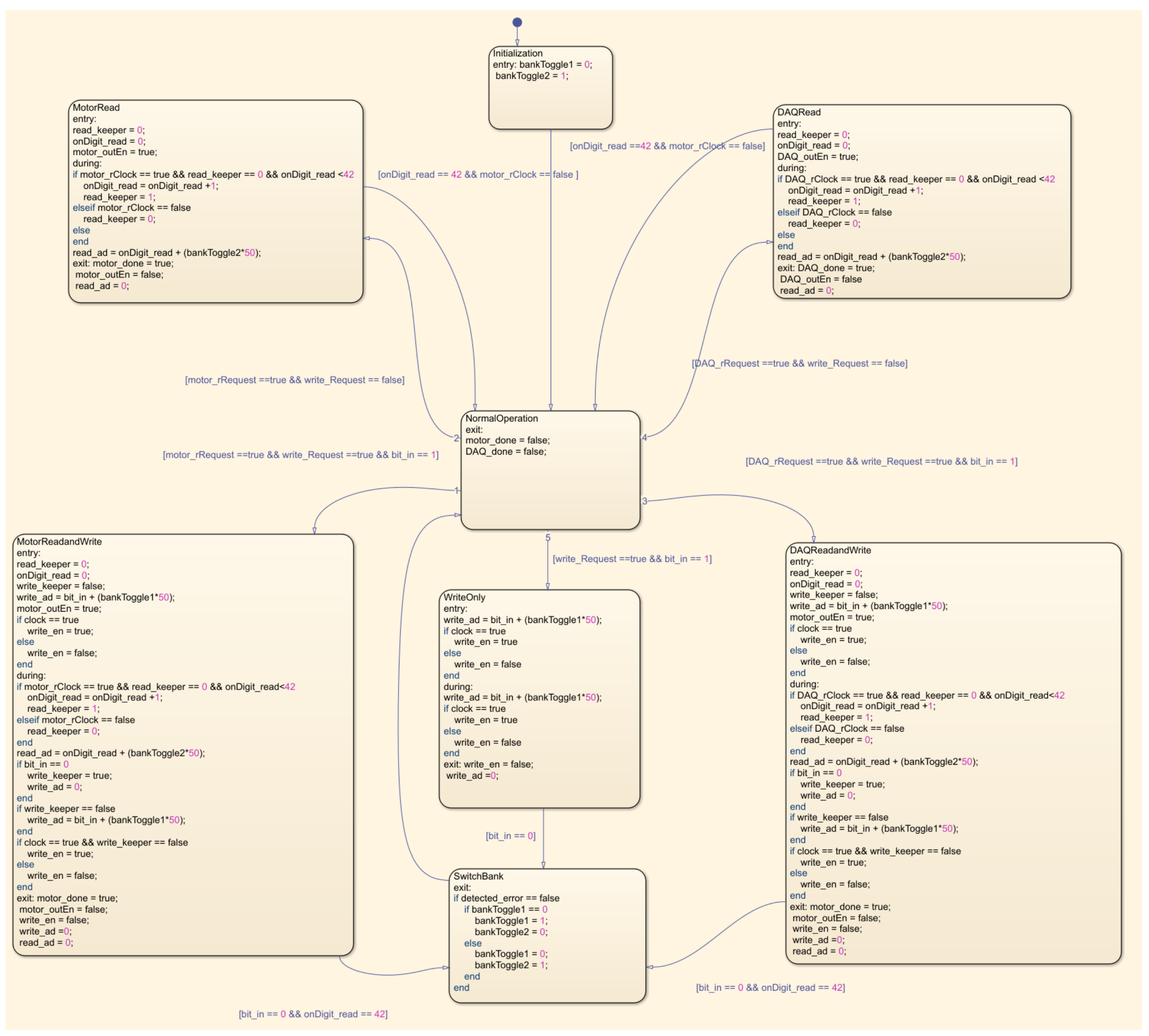


Fig. 5 – Task Scheduler



Design Continued

C Slave modules (on the right of Fig. 2) process requests from the DAQ and motor by signaling the Task Scheduler when either requests new encoder location data.

To reduce latency, read requests from the motor have the highest priority and will be executed first. If there is also a request to write new information from the encoder at the same time as a request to read information, the two events can happen simultaneously through the dual bank design: When new information from the encoder is received, it is written to bank 1. While new data is being written, the old data can be read off bank 2, to either the motor or the DAQ. Assuming that no errors are detected, the banks switch, making bank 2 the most recent data that is sent out of the device, while bank 1 is written over with new information from the encoder. If an error is detected, the banks do not switch, and the corrupt data is written over by new data before it is read.

Although this structure is not without its flaws (most noticeably, two reads cannot happen at once and when a write has started, a read must wait until the write finishes to begin) structure helps to minimize latency, while being as accurate as possible and still allowing for the encoder, motor, and DAQ to all read and write data at their native clock speeds.

Conclusions

Although the final implementation of this code is ongoing, the MATLAB simulations have shown that this splitter design is capable of quickly and accurately storing data to be shared between two different devices.

This design will be further validated through simulation of the generated VHDL code using Vivado before it will be tested on the actual encoder, motor and DAQ hardware at LCLS.

The ability for LCLS operators to log and verify the positions of mirrors on the beamline should allow for a more complete view of beamline devices.

Acknowledgments

Use of the Linac Coherent Light Source (LCLS), SLAC National Accelerator Laboratory, is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Contract No. DE-AC02-76SF00515. Additionally, this project would not have been possible without the support of my mentor Tyler Johnson.

Date: 9/10/2020